using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.ServiceModel;

using Microsoft.Crm.Sdk;

using Microsoft.Xrm.Sdk.Query;

using Microsoft.Xrm.Sdk;

using Microsoft.Crm.Sdk.Messages;

namespace UpdateAssociatedRecord

{

//Updating Child Entity Record once the Parent Entity is Updated

public class Update : IPlugin

{

//Description: Updating Jobs(selected Position Field in Candidates) from Candidates

//Message: Update

//Entity: Candidates

//Event: Post-Operation

//Image: Pre-Image, Post-Image

public void Execute(IServiceProvider serviceProvider)

{

IPluginExecutionContext context = (IPluginExecutionContext)

serviceProvider.GetService(typeof(IPluginExecutionContext));

//if (context.PreEntityImages.Contains("Target"))

//{

// Entity preMessageImage = (Entity)context.PreEntityImages["Target"];

//}

if (context.PostEntityImages.Contains("Target"))

{

Entity postMessageImage = (Entity)context.PostEntityImages["Target"];

if (postMessageImage.Attributes.Contains("weavings\_name") == true)

{

IOrganizationServiceFactory serviceFactory = (IOrganizationServiceFactory)serviceProvider.GetService(typeof(IOrganizationServiceFactory));

IOrganizationService service = serviceFactory.CreateOrganizationService(context.UserId);

EntityReference weavings\_position = postMessageImage.GetAttributeValue<EntityReference>("weavings\_position");

Entity Jobs = service.Retrieve(weavings\_position.LogicalName,weavings\_position.Id,new ColumnSet("weavings\_joblocation"));

Jobs.Attributes["weavings\_joblocation"] = postMessageImage.GetAttributeValue<string>("weavings\_location");

service.Update(Jobs);

}

}

}

}

}